Fast Forward

By Thomas Hampel (HCL) & Daniel Nashed (HCL Lifetime Ambassador)
Speakers

Daniel Nashed
- Nash!Com, Germany
- HCL Business Partner / ISV
- nsh@nashcom.de
- http://blog.nashcom.de

Thomas Hampel
- HCL Germany
- Director @ Product Management
- thomas.hampel@pnphcl.com
- https://blog.thomashampel.com

HCL Lifetime Ambassador
HCL’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice and at HCL’s sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard HCL benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.
Achtung!
Weniger als 4 Monate bis Weihnachten!

Wer sich auf Überraschungen freut ist hier in dieser Session richtig. Worum es aber genau geht bleibt noch ein Geheimnis.

Nur soviel sei verraten: Jeder Domino Administrator wird nach dieser Präsentation SEHR zufrieden sein. Wir lüften das Geheimnis exklusiv und nur hier für die Teilnehmer der DNUG – natürlich mit Live-Demo!

106 Tage bis Weihnachten
Domino Early Access Program

- Available NOW
- Free for all customers with a current entitlement

https://blog.hcltechsw.com/domino/introducing-hcl-domino-early-access-program
What is it

https://blog.hcltechsw.com/domino/introducing-hcl-domino-early-access-program

- Fast paced, technical preview to provide customers and partners the opportunity to provide feedback early in the development phase
  - Providing >specific< features in form of Code Drops
  - New code drops every couple of weeks

- Available for all existing customers with current subscription & support.

- This is **NOT** a traditional beta program!
  - There will be a separate Beta Program including Notes Clients & other server platforms at a later stage

- Public available documentation

- Discussion forum (same format we had before)
  https://registration.hclpartnerconnect.com/dominoearlyaccessforum.nsf?open

- **Docker only** with a short refresh cycle and new functionality in code drops
- Docker image download is provided via Flexnet
Docker only…

- Fast Paced
- Easier to deploy and update for an early preview!
- Containers are one of the key back-end technologies used in modern IT infrastructure
- Cloud native strategy, which leverages containers in its core
  - Cloud native does not need to mean cloud hosted!
  - The same technologies are used on-prem by many customers!

Yes we know…

- Docker only concept is a challenge for some admins
- Installers for client and server are not provided at this stage, those will be provided later.
- We are planning to have a traditional beta program where installers and language specific packages are provided – stay tuned!
Domino Early Access Program and Full Beta Plans

**Early Access Program**
- Container/Docker images only
- Purpose: feedback on specific features

**V12 Beta (Traditional Beta Programs)**
- Windows, Linux, and all platforms
- Purpose: feedback on overall release and languages

---

### Feature Overview
- Feature A
- Feature B
- Feature C
- Feature D
- Feature E
- Feature F
- Feature G

**V12 Beta 1**
- some languages

**V12 Beta 2**
- many Languages

**V12 GA**
- all languages
HCL Domino Docker Image

https://hclsoftware.flexnetoperations.com/flexnet/operationsportal/startPage.do

- Official HCL Image
- Just download from Flexnet and import to your environment = Ready to run

- https://hclsoftware.flexnetoperations.com/flexnet/operationsportal/entitledDownloadFile.action?downloadPkgId=HCL_DominoV12_Early_Access_Program&fromRecentPkg=true

HCL Domino V12 Early Access Program

<table>
<thead>
<tr>
<th>Files</th>
<th>Download Log</th>
<th>Notification Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download Selected Files</td>
<td>2 File(s)</td>
<td></td>
</tr>
</tbody>
</table>

- HCL Domino 12.0 on Docker Early Access August 2020
  - File Size: 693.04MB
  - File Added: Aug 15, 2020
  - File Name: Domino_12.0_DockerImage_FeaturePreview1.tgz
  - MDS Signature: a77e44f52b0635f441dafa3e8078739b3
  - SHA256 CheckSum: df02b7c056c4a35751278144a27ad1f2068779655591379f3488baf8b336

- HCL Domino 12.0 on Docker Early Access Release Notes
  - File Size: 112.48KB
  - File Added: Aug 14, 2020
  - File Name: early_access_release_notes.zip
Business Partner Accounts don’t have the download section

- Search for the image is a bit more complicated
- You have to look into your entitlements searching for “HCL DOMINOUTILITYSERVERValueUnit”
- Clicking “Download Now” brings you to all software part of this entitlement
- Selecting “Download Now” takes you to the download.
Basic Setup
Docker (Container) environment to use

- Currently the HCL Docker image is only officially supported on:
  RHEL / CentOS 7.4+ with Docker Engine 19+

- The underlying container services are available on many platforms
  - “Docker images” work in many environments
  - Docker, Docker Desktop, Podman, Kubernetes (K8s), OpenShift ...

- Let's have a look what is best to start with ..
Level 1 - Docker in MacOSX

- Uses Mac integrated virtualization
- Also runs a Linux VM with Container run-time
- Desktop Application
- Kubernetes single node-cluster
- Docker Compose

Start here: https://docs.docker.com/docker-for-mac/
Leverages Hyper-V
(Linux container with Container run-time)

Command Line is Windows
(only batch and power-shell scripting)

Desktop application

Kubernetes single node-cluster

Docker compose

Hyper-V and VMware did not work on the same time

- Fix from VMware and Microsoft
  You need VMware 15.5 Workstation/Player

Start here: [https://docs.docker.com/docker-for-windows/](https://docs.docker.com/docker-for-windows/)
Uses native WSL2 Linux containers for Container run-time
No requirement to run Hyper-V
Full integration in Ubuntu or Debian running in separate WSL container each
Desktop application
Kubernetes single node-cluster
Docker Compose

Start here:
https://docs.docker.com/docker-for-windows/wsl/
- WSL 2 is brand new

- Requires Windows 10 Version 2004
  - Not available for all customers yet
  - See http://blog.nashcom.de for details
Level 1 – Docker on Linux

- CentOS/RedHat, Debian, Fedora, Ubuntu (separate install instructions per platform)
- Docker service controlling + Docker Client
- Runs native on Linux
- Okay for smaller production deployments
- You can optionally install Docker Compose
- No Kubernetes included

Start here: https://docs.docker.com/engine/install/

Level 1 conclusion: Win? MacOS? Linux? It’s always Linux in one way or the other
Native Linux with Virtualization

A good starting point would be a local VM

- Well supported and tested software
  - VMware Workstation / Player / Fusion (Mac)
  - Virtual Box (Windows)
  - Most other virtualization platform works as well

- Create an empty VM for CentOS 7.x
- Get the ISO Image for CentOS 7.6 Minimum Install

- See Appendix for step by step slides

- Note: Other Linux distributions like SLES, Ubuntu will also work
Hosted Linux Machine

Example:

- Provider in Germany with modern infrastructure
  - 1 CPU, 2 GB RAM, 20 GB Disk
  - Deployed in seconds!
  - Cost < 3 Euro/Month

- Advantages
  - Static IP you can point a DNS entry to
  - No corporate firewall issues
  - Allows direct testing Let’s Encrypt HTTP challenges
- Import Image
Import Docker Image

- Import image to your Docker host via “docker” command
  - `docker load --input Domino_12.0_DockerImage_FeaturePreview1.tgz`
    - 16200ece0bc9: Loading layer [=====>] 156.1MB/156.1MB
    - aa88df1bfd40: Loading layer [=====>] 996MB/996MB
    - Loaded image: domino-docker:V1200_08042020prod

- Verify the Docker image is on your Docker server

```
docker images
REPOSITORY               TAG                  IMAGE ID            CREATED             SIZE
domino-docker            V1200_08042020prod   b67f96ca5756        13 days ago         1.35GB
```
Start Domino Server in Setup Mode

- Command-Line to create a container from an image for setup

```bash
docker run -it -p 8585:8585 \
--hostname=marvel.csi-domino.com --name domino12 \
--cap-add=SYS_PTRACE --rm \
-v notesdata:/local/notesdata \
domino-docker:V1200_08042020prod --setup
```

- Details
  - `--cap-add=SYS_PTRACE` → important for NSD / gdb
  - `-p` → publishes external ports (sets Linux firewall rules automatically!)
  - `-v` → Maps volume to the container
  - `--rm` → Deletes container on shutdown
Configure your Domino Server

- Server will start in “listen mode”
- Requires remote setup
  - No X window setup, because no X window libs are installed
  - You need to install the “Remote Server Setup tool” option on your admin client
    - Tip: Files can be also downloaded from server – see documentation below
  - It's basically the same (Java based) setup launched locally on Windows

- Documentation “Remote setup”
Plan B: Install X11 & use a remote X11 Server

- Domino setup can leverage X11 on a remote X11 server
  - Just the required X11 libs are missing
  - UBI 8 does only contain limited X11 support without Redhat subscription
  - But the contained libs are sufficient for Domino remote configuration

- Open a root bash into the running container and install missing X11 libs
  - `docker exec -it -u 0 domino12 bash`
  - `yum install -y xorg-x11-apps libX*`
  - `exit`
Install X11 & use a remote X11 Server

- Open a normal bash with “notes” user, export your DISPLAY
  - `docker exec -it -u 0 domino12 bash`
  - `export DISPLAY=192.168.96.112:0.0`

- Switch to data directory and run server process to start configuration via X11
  - `cd /local/notesdata`
  - `/opt/hcl/domino/bin/server`

- Tip: MobaXTerm on Windows is very easy to setup
  - No authorization to set. Just prompts when a remote process tries to open a session
Run your first Domino Server

- Command-Line to create a run-time container

```
docker run -it -p 80:80 -p 443:443 -p 1352:1352 \
--hostname=marvel.csi-domino.com --name domino12 \
--cap-add=SYS_PTRACE \
--stop-timeout=90 \
-v notesdata:/local/notesdata \
domino-docker:V1200_08042020prod
```

- Tips
  - In some environments with multiple IP addresses, you might need to switch to host network mode
    - `--network=host`
  - You can use “docker cp” commands to copy data from/to the container
Interact with your Domino Server

- Normal administration should be performed via Admin client
- You can attach to the Docker container to see the console and interact
  - `docker attach domino12`
  - Exit via CTRL-p + CTRL-q
- Or use a bash into the running container we used earlier for direct Linux access
  - `docker exec -it domino12 /bin/bash`
    - Return via “exit”
    - Use “-u 0” option for root access
- Remember: Software you install in a container and files you edit in the container file-system are gone, when you run a new container!
Stop a Domino Server cleanly!

- Docker by default sends a **SIGTERM** signal to the main process of the container
  - If the main process does not stop within **10 seconds**, a **SIGKILL** signal is send to the container
    - This would not allow Domino to shutdown cleanly

- **Solution**
  - Specify a longer shutdown grace period and ensure that the main process will catch the shutdown request to start the Domino shutdown
  - `docker stop --time=120 domino12`
    - Or set `--stop-timeout=90` on the run statement
Let’s Encrypt

- Request Let’s Encrypt certificates from within Domino
- Free of charge SSL/TLS certificates
- No need to copy files to server
- No need to create or manage *.kyr files
Code Drop #1

Let's Encrypt

Trusted Certificate

https://yourserver.com

HCL Domino V12
Automatic Integrated Certificate Management

- First early code drop to get feedback

- Currently implemented
  - Let's Encrypt® certificate requests leveraging “HTTP” challenges
  - New Domino Server task “certmgr” and database “certstore.nsf”

- Certificates and keys are stored in PEM format
  - Kyr-File format is still required (for now)
    - will remain as an option for older servers

- Basic flows are implemented
  - Will evolve over code drops – also based on your feedback!
Quick Start:
- Domino Configuration
- Request & Install Let’s Encrypt Certificate
Start the Certificate Manager Task on the Domino server

- `load certmgr -ACCEPT_TOU_AUTO_CONFIG`

  - This command is equivalent to:
  `load certmgr -r -c -o -y -ACCEPT_TOU`

What if your Domino server is hosting more than one internet site?, e.g.:
  - domino.csi-domino.com
  - dnug.csi-domino.com

Requesting multiple Subject Alternate Names

Key rollover from RSA keys to (new) ECDSA keys

Technology

- Leverages existing and new Domino security APIs

- Implements Let's Encrypt uses ACME protocol V2 (RFC 8555)
  - **AMCE = Automatic Certificate Management Environment**
  - Own HCL implementation leveraging standards like
    - JSON, LibCurl, JWS, OpenSSL, Notes crypto including PEM, RSA and ECDSA keys ...

- Designed for automation
  - When server is available on HTTP (port 80) and HTTPS (port 443) **and** has an DNS entry, you can create your first certificate with one command today
Architecture Overview

Components

*) Connection between Domino, LE and CertStore could be local or NRPC
Domino HTTP and LE could be on separate server and just need a common CertStore.

(A) X.509/KYR
(B) Challenge needed to verify request
(C) LE used to authenticate with ACME CA
(D) Proxy account needed for outgoing communication

Flow

1. LE creates account (C) with ACME server
2. LE creates private key and writes it to CertStore (A)
3. LE creates CSR and sends it to ACME CA (*)
4. LE puts received challenge (B) in CertStore
5. ACME server requests challenge on port 80 to verify
6. Domino HTTP replies with challenge (B) from CertStore
7. LE receives certificate including and writes it to CertStore (A)

HTTP (and INET tasks) read X.509 from CertStore (A)

*) Proxy communication uses Proxy user (D)
Standard Scenario - Domino V12 with inbound HTTP(S)

- Certmgr task and certstore.nsf on the same Domino V12 server
- DNS pointing to your server
  - DNS Domain like *.csi-domino.com pointing to the server
  - Allows to test with any host name ;-)
- Certmgr supports outbound proxy connections including authentication
- Inbound HTTP(S) connection can use an incoming proxy/load-balancer
- Allows automatic setup if DNS and hostname matches
Distributed Setup

- Local Domino V12 server (e.g. on notebook)
  - Notes.ini `certmgr_server=domino-v11-server` pointing to an existing server with inbound HTTP(S)
- Domino V10/11 server running in corporate environment/at provider
  - DSAPI installed + certstore.nsf on Domino V11 server handling incoming Let's Encrypt Challenges
    - DSAPI is available for Linux64 and Win64 and works on Domino 10,11 & 12
- Certmgr running locally can request certificates for remote server
  - .kyr file has to be deployed manually on remote server
- You could also point your existing server to the Domino V12
  - If inbound NRPC is allowed
  - Notes.ini `certmgr_server=domino-v12-server` pointing to your internal server
What's Next?
What if you don’t want or can’t use Let’s Encrypt?

Creating Certificate Request Manually
- Support for (just any) Certificate Authority

→ DEMO #4: Request Certificate manually from a local Microsoft Certificate Authority
Coming Soon:

- New “Code Drops” every couple of weeks

- Candidates for next Code Drop:
  - AES Encryption for DAOS
  - Replicate Quota Settings
  - Pubnames Template enhancements
  - TOTP
Time-based One-time Password (TOTP)

- Two-Factor Authentication built into Domino
- App-Based (e.g. Google Authenticator, Authy, etc.)

Coming Soon:

[Link to Aha Idea: https://domino.ideas.aha.io/ideas/DOMINO-I-29]
Q & A
Questions & Keep the Ideas Coming – Domino Ideas Portal

And, please follow these three steps:

1. Where we gather and prioritize your ideas
   - Search the forum to see if your idea already exists.
   - Vote for any ideas and add any additional comments that you may have related to that idea.

2. Where we want you to add, vote, comment on and share as many ideas as you like.
   - If your idea does not already exist, create a new idea and format your idea like: “As a <insert role>, I would like to be able to <insert action> in order to <insert need>.”

3. Create as many ideas as you’d like, but remember to keep each idea, separate.
   - Do not create idea that consist of more than one request for enhancement.

https://domino.ideas.aha.io/
$8.4 BILLION ENTERPRISE | 132,000 IDEAPRENEURS | 44 COUNTRIES

WATCH THE FILM
BACKUP
**What’s Included**
- HCL Domino
- HCL Verse
- HCL Traveler
- HCL Domino Appdev Pack
- HCL Domino Volt
- HCL Notes
- HCL Domino Designer

**What is it**
- Docker image available for *time limited* trial of Domino products
- Available via direct and partner channels

**What it isn’t**
- This is not a self service trial directly available to customers.

**How does it work**
- Trials are initiated via an order in CRM
- The order will trigger the generation of a time limited (30 day) licence key
  - Trials can be extended with the request of additional keys
- A welcome letter will be generated with the key and installation instructions and sent to the customer
- Customers will have access to the code via Flexnet upon order submission

[https://help.hcltechsw.com/domino/11.0.1/admin/trial_over.html](https://help.hcltechsw.com/domino/11.0.1/admin/trial_over.html)
Create a Virtual Machine with VMWare Workstation
Virtual Machine Setup – New VM

Welcome to the New Virtual Machine Wizard

What type of configuration do you want?

- Typical (recommended)
  Create a Workstation 14.x virtual machine in a few easy steps.

- Custom (advanced)
  Create a virtual machine with advanced options, such as a SCSI controller type, virtual disk type and compatibility with older VMware products.
Virtual Machine Setup – Hardware

- Define/Confirm Hardware configuration
Virtual Machine Setup – Select ISO

- Choose CentOS Image that was previously downloaded
Virtual Machine Setup – Name & Location

Define

- Name of your new virtual Machine
- Location of where the VM should be located
Virtual Machine Setup – CPU Settings

- Define # of CPUs & CPU Cores

- For testing 2 cores are enough.
Virtual Machine Setup – RAM Settings

- Define allocated memory / RAM
- For testing 2 Gbyte are enough
For network connection type choose “NAT”
Virtual Machine Setup – Disk Controller

- Keep the defaults
Virtual Machine Setup – Disk Type

- Keep the defaults
Create a new virtual disk that will serve as disk for the VM.
Define a file name for the virtual disk
Virtual Machine Setup – Disk Storage

- Size: 20 GB
- Choose “Store virtual disk as a single file”
Virtual Machine Setup – Installation Summary

- Review settings + click Finish
Install CentOS from Minimum ISO
Choose “Install CentOS 7”
Booting up

- Press the <ENTER> key to begin the installation process.

    7.9652821 dracut-pre-udev[3341]: modprobe: ERROR: could not insert 'floppy':
    No such device
    [ OK ] Started Show Plymouth Boot Screen.
    [ OK ] Started Forward Password Requests to Plymouth Directory Watch.
    [ OK ] Reached target Paths.
    [ OK ] Reached target Basic System.
    [ OK ] Started Device-Mapper Multipath Device Controller.
    Starting Open-iscsi...
    [ OK ] Started Open-iscsi.
    Starting dracut inittab hook...
    Mounting Configuration File System...
    [ OK ] Started Show Plymouth Boot Screen.
    [ OK ] Started Forward Password Requests to Plymouth Directory Watch.
    [ OK ] Reached target Paths.
    [ OK ] Reached target Basic System.
    [ OK ] Started Device-Mapper Multipath Device Controller.
    Starting Open-iscsi...
    [ OK ] Started Open-iscsi.
    Starting dracut inittab hook...
    Mounting Configuration File System...
    [ OK ] Dracut mount: /dev/sr0 is write-protected, mounting read-only
    [ OK ] Created slice system-checkiso9953-5.mount.
    Starting Media check on /dev/sr0...
    /dev/sr0: e2fsck: +9662e962e5d263b49f1fe1326e826
    Fragment sums: 2a3f135f1bc99a9e90f77e1bb76a794fbc6de8e11b1543e35eb14e9a1
    Fragment count: 20
    Press [Esc] to abort check.
    Checking: 049.4%
CentOS Installation – Select Language

- Select preferred language
Button “Begin Installation” is greyed out until all required parameters are defined.
Optional:

- Add your preferred Keyboard Layout
CentOS Installation – Select Install Disk

- Choose destination volume
Network Configuration
CentOS Installation – Timezone Configuration

- Define Timezone
CentOS Installation – Begin Installation

- Click “Begin Installation”
While installation is running, define your root password
CentOS Installation – Choose a “secure” Password

- Make sure the PW is using a sufficient quality (length, complexity, entropy)
- Remember this PW, you’ll need it later on.
CentOS Installation – Done

- DONE !!!
CentOS Linux 7 (Core)
Kernel 3.10.0-957.el7.x86_64 on an x86_64

localhost login: root
Password:
Last login: Tue Apr  9 15:09:22 on tty1
[root@localhost ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
   link/ether 00:0c:29:bf:ce:0c brd ff:ff:ff:ff:ff:ff
   inet 192.168.100.201/24 brd 192.168.100.255 scope global noprefixroute dynamic ens33
       valid_lft 1718sec preferred_lft 1718sec
   inet6 fe80::6c2a:fffe:fe9f:d8f/64 scope global noprefixroute
data1:4ba5:5a2b:1800:1b25:fffe:fe9f:d8f/64 scope global noprefixroute dynamic
       valid_lft 86386sec preferred_lft 14306sec
   inet6 fe00::57ba:8dc3:ac0:47ff/64 scope link noprefixroute
data2:4ba5:5a2b:1800:1b25:fffe:fe9f:d8f/64 scope link noprefixroute
data3:4ba5:5a2b:1800:1b25:fffe:fe9f:d8f/64 scope link noprefixroute
data4:4ba5:5a2b:1800:1b25:fffe:fe9f:d8f/64 scope link noprefixroute
   [root@localhost ~]#
Get Servers IP Address

- Minimum ISO install is very basic and has almost no tools installed
  - “ifconfig” is not installed yet
- Get server's IP address via “ip addr”

```bash
ip addr
```

```
1: enol6777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
   link/ether 00:0c:29:38:15:77 brd ff:ff:ff:ff:ff:ff
   inet 192.168.96.100/24 brd 192.168.100.255 scope global noprefixroute enol6777736
      valid_lft forever preferred_lft forever
```
Remote Login via Putty
Docker Installation
Great online documentation
- https://docs.docker.com
- Reference for all commands
- Documentation for installation per platform

The next slides cover the basic concepts and installation

Side note: If you install CentOS 8 it might automatically install “podman”
- Podman is the featured new container environment RedHat supports
  - It is more or less Docker compatible
  - The commands are very similar and there is a compatibility mode to use Docker commands
  - If you are first looking into Docker/Containers you should stay on CentOS 7!
CentOS – Required Packages before installing Docker

- “yum” is used to install software packages on CentOS
- `yum install -y net-tools wget yum-utils device-mapper-persistent-data lvm2`
  - `-y` means yes to prompts
  - net-tools, yum-utils, wget and net-tools → important tools
  - device-mapper-persistent-data, lvm2 → required by Docker
- `yum update`
  - Updates existing packages and kernel to the current version
- Docker has its own software repository
  - `yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo`
    - Adds the docker repository for software download
Install Docker CE

- Docker CE is a free **Community Edition**
- Install Docker CE, the command-line and containerd
  - `yum install -y docker-ce docker-ce-cli containerd.io`
- Allow this host to forward/route IP traffic and restart the network
  - `echo net.ipv4.ip_forward=1 >> /etc/sysctl.conf`
  - `systemctl restart network`
  - Required because Docker uses its own network interfaces → else no outside communication
- Enable (auto start) and start the Docker Service
  - `systemctl start docker`
  - `systemctl enable docker`
Run the Docker Hello World image to verify the Docker installations is OK
  - `docker run hello-world`

Optional: Run CentOS latest interactive (container is removed afterwards)
  - `docker run --rm -it centos:latest bash`

In both cases Docker pulls down the image from Docker registry/hub, creates a new container and runs it
Work with Docker

- **Important commands**
  - `docker images` → lists all locally available images
  - `docker ps` → shows all running containers
  - `docker ps -a` → also shows stopped containers
  - `docker run` → creates container from image and starts it
  - `docker start/stop` → starts/stops existing containers
  - `docker inspect` → shows detailed information for a container
  - `docker volume ls` → lists existing volumes
  - `docker volume rm` → removes a volume
  - `docker exec` → executes a command inside a container
Docker Volumes

- By default all data is stored in the container
  - For applications with local storage requirements like Domino, this does not work well

- Docker supports "volumes" which are "mapped" in to the container
  - The data from the local directory will be copied to the volume at first run when the volume is empty

- The default implementation is a local disk
  - You can either create a volume manually, mount existing directories or let Docker create it

[https://docs.docker.com/storage/volumes/]
Docker Volumes

- **Multiple options**
  - You can specify an existing volume in your run statement
    - `docker run --rm -it -v /local/data1:/local/data` centos:latest bash
  - Or let Docker create a local volume
    - `docker run --rm -it -v test-data1:/local/data` centos:latest bash
    - Default location: `/var/lib/docker/volumes`
  - Or for example use a NFS mount on a NAS
    - `docker volume create --driver local --opt type=nfs --opt o=addr=192.168.101.42,rw --opt device=:/data/docker_vol --name nfsvol`
    - `docker run --rm -it -v nfsvol:/local` centos:latest bash
Docker Volume Commands

- **docker volume ls**
  - Lists all volumes

- **docker volume inspect my-vol**
  - Shows details about one volume

- **docker volume create my-vol**
  - Creates local volume

- **docker volume rm my-vol**
  - Removes volume!

- **docker system df**
  - Checks for used/free space

- **docker system prune**
  - WARNING! This will remove:
    - all stopped containers
    - all networks not used by at least one container
    - all dangling images
    - all dangling build cache